

## SYSTEM AND METHOD FOR ON-BOARD DIAGNOSTICS OF MEMORY MODULES

### TECHNICAL FIELD

The present invention relates to a computer system, and more particularly, to  
5 a computer system having a memory module with a memory hub coupling several memory  
devices to a processor or other memory access devices.

### BACKGROUND OF THE INVENTION

Computer systems use memory devices, such as dynamic random access  
memory ("DRAM") devices, to store instructions and data that are accessed by a processor.  
10 These memory devices are normally used as system memory in a computer system. In a  
typical computer system, the processor communicates with the system memory through a  
processor bus and a memory controller. The processor issues a memory request, which  
includes a memory command, such as a read command, and an address designating the  
location from which data or instructions are to be read. The memory controller uses the  
15 command and address to generate appropriate command signals as well as row and column  
addresses, which are applied to the system memory. In response to the commands and  
addresses, data is transferred between the system memory and the processor. The memory  
controller is often part of a system controller, which also includes bus bridge circuitry for  
coupling the processor bus to an expansion bus, such as a PCI bus.

20 Although the operating speed of memory devices has continuously  
increased, this increase in operating speed has not kept pace with increases in the operating  
speed of processors. Even slower has been the increase in operating speed of memory  
controllers coupling processors to memory devices. The relatively slow speed of memory  
controllers and memory devices limits the data bandwidth between the processor and the  
25 memory devices.

In addition to the limited bandwidth between processors and memory devices, the performance of computer systems is also limited by latency problems that increase the time required to read data from system memory devices. More specifically, when a memory device read command is coupled to a system memory device, such as a synchronous DRAM ("SDRAM") device, the read data are output from the SDRAM device only after a delay of several clock periods. Therefore, although SDRAM devices can synchronously output burst data at a high data rate, the delay in initially providing the data can significantly slow the operating speed of a computer system using such SDRAM devices.

One approach to alleviating the memory latency and bandwidth problem is to use multiple memory devices coupled to the processor through a memory hub. In a memory hub architecture, a system controller or memory hub controller is coupled to several memory modules, each of which includes a memory hub coupled to several memory devices. The memory hub efficiently routes memory requests and responses between the controller and the memory devices. Computer systems employing this architecture can have a higher bandwidth because a processor can access one memory device while another memory device is responding to a prior memory access. For example, the processor can output write data to one of the memory devices in the system while another memory device in the system is preparing to provide read data to the processor. The operating efficiency of computer systems using a memory hub architecture can make it more practical to vastly increase memory capacity in computer systems.

Although there are advantages to utilizing a memory hub for accessing memory devices, the design of the hub memory system, and more generally, computer systems including such a memory hub architecture, becomes increasingly difficult. For example, in many hub based memory systems, the processor is coupled to the memory via a high speed bus or link over which signals, such as command, address, or data signals, are transferred at a very high rate. However, as transfer rates increase, the time for which a signal represents valid information is decreasing. As commonly referenced by those

ordinarily skilled in the art, the window or “eye” for the signals decreases at higher transfer rates. With specific reference to data signals, the “data eye” decreases. As understood by one skilled in the art, the data eye for each of the data signals defines the actual duration that each signal is valid after various factors affecting the signal are considered, such as  
5 timing skew, voltage and current drive capability, and the like. In the case of timing skew of signals, it often arises from a variety of timing errors such as loading on the lines of the bus and the physical lengths of such lines.

As data eyes of the signals decrease at higher transfer rates, it is possible that one or more of a group of signals provided in parallel will have arrival times such that not  
10 all signals are simultaneously valid at a receiving entity, and thus cannot be successfully captured by that entity. For example, where a plurality of signals are provided in parallel over a bus, the data eye of one or more of the particular signals do not overlap with the data eyes of the other signals. In this situation, the signals having non-overlapping data eyes are not valid at the same time as the rest of the signals, and consequently, cannot be  
15 successfully captured by the receiving entity.

Clearly, as those ordinarily skilled in the art will recognize, the previously described situation is unacceptable. As it is further recognized by those familiar in the art of high speed digital systems, signal timing and signal integrity are issues that have become increasingly more significant in the design of systems capable of transferring and  
20 transmitting information at high speeds because signal characteristics can be affected by many things. As a result, diagnostic analysis and evaluation of signals, whether command, address, or data signals, is becoming a more critical step in the design process for any high-speed digital system. Examples of the types of issues evaluated through diagnostic testing include pattern sensitivity, power and ground sensitivity, voltage margin, signal interactions  
25 on a bus, failure analysis, and the like.

The tools typically used in performing diagnostics include logic analyzers, pattern generators, oscilloscopes, and in some cases, modified desktop computers. It will be appreciated that there are many other diagnostic tools that are available, however, one

common feature shared by all of these tools is the relatively expensive cost. In many instances, only well funded companies can afford equipment with enough sophistication capable of performing diagnostics on high-speed systems. Often, smaller, less well funded companies must compromise performance of the diagnostic equipment in order to afford  
5 the equipment, thus, either making some diagnostic evaluation more difficult, or perhaps, even impossible.

Another issue that often arises with conventional diagnostic tools is the manner in which signals are detected by the diagnostic equipment. More specifically, probes of various sorts are used to couple signals from a signal line for detection by the  
10 diagnostic equipment. A problem resulting from this is that the probe can introduce loading effects that change the characteristic of the signal being evaluated. Although probes are specifically designed to have high impedance and low capacitance to minimize loading issues and the introduction of noise, there is still in many cases, an unacceptable level of loading that changes the character of a signal to such a degree that it cannot be  
15 accurately evaluated.

Another issue that is specific to performing diagnostics on a memory system is the difficulty associated with obtaining control over the memory bus in order to perform evaluation. The ability to evaluate a memory system often requires that specific signals of interest can be captured and analyzed by obtaining control of the memory bus and  
20 monitoring the interaction of the signal with the bus itself. Unless control over the memory bus can be obtained, analysis becomes a difficult task. However, obtaining control over the memory bus is a difficult task in itself because conventional approaches often interfere with the normal operation of the computer system, thus, preventing accurate analysis of the memory system under true, normal operating conditions.

25 Therefore, there is a need for alternative approaches to performing diagnostic analysis and evaluation on memory modules, including those memory modules for use in a memory hub architecture.

## SUMMARY OF THE INVENTION

The present invention is directed to a memory hub having an on-board diagnostic engine through which diagnostic testing and evaluation of the memory system can be performed. The memory hub includes a link interface for receiving memory requests for access to memory devices of the memory system and a memory device interface coupled to the memory devices for coupling memory requests to the memory devices for access to at least one of the memory devices. A switch for selectively coupling the link interface and the memory device interface is further included, and a memory hub diagnostic engine is coupled to the switch for coupling control signals to the link interface and the memory device interface to perform diagnostic testing of the memory system. The diagnostic engine includes a maintenance port that provides access to results of the diagnostic testing and through which diagnostic testing commands can be received.

## BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a computer system having memory modules in a memory hub architecture in which embodiments of the present invention can be implemented.

Figure 2 is a block diagram of a memory hub according to an embodiment of the present invention for use with the memory modules in the computer system of Figure 1.

Figure 3 is a block diagram of a diagnostic engine according to an embodiment of the present invention.

## DETAILED DESCRIPTION OF THE INVENTION

Embodiments of the present invention are directed to a system memory having memory modules with on-board diagnostics and self-testing capability. Certain details are set forth below to provide a sufficient understanding of the invention. However, it will be clear to one skilled in the art that the invention may be practiced without these particular details. In other instances, well-known circuits, control signals, and timing

protocols have not been shown in detail in order to avoid unnecessarily obscuring the invention.

A computer system 100 according to one embodiment of the invention is shown in Figure 1. The computer system 100 includes a processor 104 for performing  
5 various computing functions, such as executing specific software to perform specific calculations or tasks. The processor 104 includes a processor bus 106 that normally includes an address bus, a control bus, and a data bus. The processor bus 106 is typically coupled to cache memory 108, which, is typically static random access memory ("SRAM"). Finally, the processor bus 106 is coupled to a system controller 110, which is also  
10 sometimes referred to as a "North Bridge" or "memory controller."

The system controller 110 serves as a communications path to the processor 104 for a variety of other components. More specifically, the system controller 110 includes a graphics port that is typically coupled to a graphics controller 112, which is, in turn, coupled to a video terminal 114. The system controller 110 is also coupled to one or  
15 more input devices 118, such as a keyboard or a mouse, to allow an operator to interface with the computer system 100. Typically, the computer system 100 also includes one or more output devices 120, such as a printer, coupled to the processor 104 through the system controller 110. One or more data storage devices 124 are also typically coupled to the processor 104 through the system controller 110 to allow the processor 104 to store data or  
20 retrieve data from internal or external storage media (not shown). Examples of typical storage devices 124 include hard and floppy disks, tape cassettes, and compact disk read-only memories (CD-ROMs).

The system controller 110 includes a memory hub controller 128 that is coupled to several memory modules 130a,b...n, which serve as system memory for the  
25 computer system 100. The memory modules 130 are preferably coupled to the memory hub controller 128 through a high-speed link 134, which may be an optical or electrical communication path or some other type of communications path. In the event the high-speed link 134 is implemented as an optical communication path, the optical

communication path may be in the form of one or more optical fibers. In such case, the memory hub controller 128 and the memory modules will include an optical input/output port or separate input and output ports coupled to the optical communication path. The memory modules 130 are shown coupled to the memory hub controller 128 in a point-to-point arrangement in which the single high-speed link 134 is coupled to all of the memory modules 130. However, it will be understood that other topologies may also be used, such as a point-to-point coupling arrangement in which a separate high-speed link (not shown) is used to couple each of the memory modules 130 to the memory hub controller 128. A switching topology may also be used in which the memory hub controller 128 is selectively coupled to each of the memory modules 130 through a switch (not shown). Other topologies that may be used will be apparent to one skilled in the art.

Each of the memory modules 130 includes a memory hub 140 for controlling access to a plurality of memory devices 148, which, in the example illustrated in Figure 1, are synchronous dynamic random access memory ("SDRAM") devices. Although shown in Figure 1 as having six memory devices 148, a fewer or greater number of memory devices 148 may be used, and memory devices other than SDRAM devices may also be used. For example, in an alternative embodiment of the present invention, a memory module includes a memory hub for controlling between nine and eighteen memory devices. The memory hub 140 is coupled to each of the system memory devices 148 through a bus system 150, which normally includes a control bus, an address bus, and a data bus.

A memory hub 200 according to an embodiment of the present invention is shown in Figure 2. The memory hub 200 can be substituted for the memory hub 140 of Figure 1. The memory hub 200 is shown in Figure 2 as being coupled to four memory devices 240a-d, which, in the present example are conventional SDRAM devices. Examples of conventional SDRAM devices include multiple data rate memory devices, such as double data rate (DDR) devices, DDR II and DDR III devices, and the like. In an alternative embodiment, the memory hub 200 is coupled to four different banks of memory

devices, rather than merely four different memory devices 240a-d, with each bank typically having a plurality of memory devices. However, for the purpose of providing an example, the present description will be with reference to the memory hub 200 coupled to the four memory devices 240a-d. It will be appreciated that the necessary modifications to the  
5 memory hub 200 to accommodate multiple banks of memory is within the knowledge of those ordinarily skilled in the art.

Further included in the memory hub 200 are link interfaces 210a-d and 212a-d for coupling the memory module on which the memory hub 200 is located to a first high speed data link 220 and a second high speed data link 222, respectively. As  
10 previously discussed with respect to Figure 1, the high speed data links 220, 222 can be implemented using an optical or electrical communication path or some other type of communication path. The link interfaces 210a-d, 212a-d are conventional, and include circuitry used for transferring data, command, and address information to and from the high speed data links 220, 222. As well known, such circuitry includes transmitter and receiver  
15 logic known in the art. It will be appreciated that those ordinarily skilled in the art have sufficient understanding to modify the link interfaces 210a-d, 212a-d to be used with specific types of communication paths, and that such modifications to the link interfaces 210a-d, 212a-d can be made without departing from the scope of the present invention. For example, in the event the high-speed data link 220, 222 is implemented using an optical  
20 communications path, the link interfaces 210a-d, 212a-d will include an optical input/output port that can convert optical signals coupled through the optical communications path into electrical signals.

The link interfaces 210a-d, 212a-d are coupled to the a switch 260 through a plurality of bus and signal lines, represented by busses 214. The busses 214 are  
25 conventional, and include a write data bus and a read data bus, although a single bi-directional data bus may alternatively be provided to couple data in both directions through the link interfaces 210a-d, 212a-d. It will be appreciated by those ordinarily skilled in the art that the busses 214 are provided by way of example, and that the busses 214 may



include fewer or greater signal lines, such as further including a request line and a snoop line, which can be used for maintaining cache coherency.

The link interfaces 210a-d, 212a-d include circuitry that allow the memory hub 140 to be connected in the system memory in a variety of configurations. For example, the point-to-point arrangement, as shown in Figure 1, can be implemented by coupling each memory module to the memory hub controller 128 through either the link interfaces 210a-d or 212a-d. Alternatively, another configuration can be implemented by coupling the memory modules in series. For example, the link interfaces 210a-d can be used to couple a first memory module and the link interfaces 212a-d can be used to couple a second memory module. The memory module coupled to a processor, or system controller, will be coupled thereto through one set of the link interfaces and further coupled to another memory module through the other set of link interfaces. In one embodiment of the present invention, the memory hub 200 of a memory module is coupled to the processor in an arrangement in which there are no other devices coupled to the connection between the processor 104 and the memory hub 200. This type of interconnection provides better signal coupling between the processor 104 and the memory hub 200 for several reasons, including relatively low capacitance, relatively few line discontinuities to reflect signals and relatively short signal paths.

The switch 260 is further coupled to four memory interfaces 270a-d which are, in turn, coupled to the system memory devices 240a-d, respectively. By providing a separate and independent memory interface 270a-d for each system memory device 240a-d, respectively, the memory hub 200 avoids bus or memory bank conflicts that typically occur with single channel memory architectures. The switch 260 is coupled to each memory interface through a plurality of bus and signal lines, represented by busses 274. The busses 274 include a write data bus, a read data bus, and a request line. However, it will be understood that a single bi-directional data bus may alternatively be used instead of a separate write data bus and read data bus. Moreover, the busses 274 can include a greater or lesser number of signal lines than those previously described.

In an embodiment of the present invention, each memory interface 270a-d is specially adapted to the system memory devices 240a-d to which it is coupled. More specifically, each memory interface 270a-d is specially adapted to provide and receive the specific signals received and generated, respectively, by the system memory device 240a-d to which it is coupled. Also, the memory interfaces 270a-d are capable of operating with system memory devices 240a-d operating at different clock frequencies. As a result, the memory interfaces 270a-d isolate the processor 104 from changes that may occur at the interface between the memory hub 230 and memory devices 240a-d coupled to the memory hub 200, and it provides a more controlled environment to which the memory devices 240a-d may interface.

The switch 260 coupling the link interfaces 210a-d, 212a-d and the memory interfaces 270a-d can be any of a variety of conventional or hereinafter developed switches. For example, the switch 260 may be a cross-bar switch that can simultaneously couple link interfaces 210a-d, 212a-d and the memory interfaces 270a-d to each other in a variety of arrangements. The switch 260 can also be a set of multiplexers that do not provide the same level of connectivity as a cross-bar switch but nevertheless can couple the some or all of the link interfaces 210a-d, 212a-d to each of the memory interfaces 270a-d. The switch 260 may also includes arbitration logic (not shown) to determine which memory accesses should receive priority over other memory accesses. Bus arbitration performing this function is well known to one skilled in the art.

With further reference to Figure 2, each of the memory interfaces 270a-d includes a respective memory controller 280, a respective write buffer 282, and a respective cache memory unit 284. The memory controller 280 performs the same functions as a conventional memory controller by providing control, address and data signals to the system memory device 240a-d to which it is coupled and receiving data signals from the system memory device 240a-d to which it is coupled. The write buffer 282 and the cache memory unit 284 include the normal components of a buffer and cache memory, including a tag memory, a data memory, a comparator, and the like, as is well known in the art. The

memory devices used in the write buffer 282 and the cache memory unit 284 may be either DRAM devices, static random access memory ("SRAM") devices, other types of memory devices, or a combination of all three. Furthermore, any or all of these memory devices as well as the other components used in the cache memory unit 284 may be either embedded  
5 or stand-alone devices.

The write buffer 282 in each memory interface 270a-d is used to store write requests while a read request is being serviced. In a such a system, the processor 104 can issue a write request to a system memory device 240a-d even if the memory device to which the write request is directed is busy servicing a prior write or read request. Using  
10 this approach, memory requests can be serviced out of order since an earlier write request can be stored in the write buffer 282 while a subsequent read request is being serviced. The ability to buffer write requests to allow a read request to be serviced can greatly reduce memory read latency since read requests can be given first priority regardless of their chronological order. For example, a series of write requests interspersed with read requests  
15 can be stored in the write buffer 282 to allow the read requests to be serviced in a pipelined manner followed by servicing the stored write requests in a pipelined manner. As a result, lengthy settling times between coupling write request to the memory devices 270a-d and subsequently coupling read request to the memory devices 270a-d for alternating write and read requests can be avoided.

20 The use of the cache memory unit 284 in each memory interface 270a-d allows the processor 104 to receive data responsive to a read command directed to a respective system memory device 240a-d without waiting for the memory device 240a-d to provide such data in the event that the data was recently read from or written to that memory device 240a-d. The cache memory unit 284 thus reduces the read latency of the  
25 system memory devices 240a-d to maximize the memory bandwidth of the computer system. Similarly, the processor 104 can store write data in the cache memory unit 284 and then perform other functions while the memory controller 280 in the same memory

interface 270a-d transfers the write data from the cache memory unit 284 to the system memory device 240a-d to which it is coupled.

Further included in the memory hub 200 is a built in self-test (BIST) and diagnostic engine 290 coupled to the switch 260 through a diagnostic bus 292. The diagnostic engine 290 is further coupled to a maintenance bus 296, such as a System Management Bus (SMBus) or a maintenance bus according to the Joint Test Action Group (JTAG) and IEEE 1149.1 standards. Both the SMBus and JTAG standards are well known by those ordinarily skilled in the art. Generally, the maintenance bus 296 provides a user access to the diagnostic engine 290 in order to perform memory channel and link diagnostics. For example, the user can couple a separate PC host via the maintenance bus 296 to conduct diagnostic testing or monitor memory system operation. By using the maintenance bus 296 to access diagnostic test results, issues related to the use of test probes, as previously discussed, can be avoided. It will be appreciated that the maintenance bus 296 can be modified from conventional bus standards without departing from the scope of the present invention. It will be further appreciated that the diagnostic engine 290 should accommodate the standards of the maintenance bus 296, where such a standard maintenance bus is employed. For example, the diagnostic engine should have a maintenance bus interface compliant with the JTAG bus standard where such a maintenance bus is used.

Further included in the memory hub 200 is a DMA engine 286 coupled to the switch 260 through a bus 288. The DMA engine 286 enables the memory hub 200 to move blocks of data from one location in the system memory to another location in the system memory without intervention from the processor 104. The bus 288 includes a plurality of conventional bus lines and signal lines, such as address, control, data busses, and the like, for handling data transfers in the system memory. Conventional DMA operations well known by those ordinarily skilled in the art can be implemented by the DMA engine 286. A more detailed description of a suitable DMA engine can be found in commonly assigned, co-pending U.S. Patent Application No. \_\_\_\_\_, entitled

APPARATUS AND METHOD FOR DIRECT MEMORY ACCESS IN A HUB-BASED MEMORY SYSTEM, filed on \_\_\_\_\_, which is incorporated herein by reference. As described in more detail in the aforementioned patent application, the DMA engine 286 is able to read a link list in the system memory to execute the DMA memory operations without processor intervention, thus, freeing the processor 104 and the bandwidth limited system bus from executing the memory operations. The DMA engine 286 can also include circuitry to accommodate DMA operations on multiple channels, for example, for each of the system memory devices 240a-d. Such multiple channel DMA engines are well known in the art and can be implemented using conventional technologies.

10           The diagnostic engine 290 and the DMA engine 286 are preferably embedded circuits in the memory hub 200. However, including separate a diagnostic engine and a separate DMA device coupled to the memory hub 200 is also within the scope of the present invention.

Embodiments of the present invention provide an environment for investigating new memory interface and high speed link technology, such as that previously discussed with respect to Figures 1 and 2. Embodiments of the present invention can be further used to provide continuous reliability data, as well as gathering error rate or margin data which can be evaluated by the host in determining appropriate action to be taken. This environment creates a rapid proto-typing capability for new memory technology.

20   Diagnostic evaluation of the memory system can be performed and monitored at the overall system level, as well as at the memory module level. That is, in addition to evaluating overall memory system performance, access to each individual memory module can be made through the respective maintenance bus to perform diagnostic testing of each memory module, thus, providing information for individual memory module performance.

25           The diagnostic engine 290, link interfaces 210a-d, 212a-d, the DMA engine 286, the maintenance bus 296 and hardware instrumentation provide a fully automated memory module test vehicle. In embodiments of the present invention, a user is able to access pattern generation and interface synchronization logic contained in the memory hub

200 through the diagnostic engine 290 to perform automated testing and diagnostic evaluation of memory device address eyes, data eyes, voltage margin, high speed hub-to-hub (link) interfaces, and boundary scans. For example, the memory modules 130 (Figure 1) can be directed to execute built in self-tests upon power up of the computer system 100.

5 The results of the self-test can be accessed through the maintenance bus 296 and the diagnostic engine 290. Additionally, further testing or diagnostics can be performed by user access of the diagnostic engine 290 through the maintenance bus 296 to further investigate any issues. The system under test, or a separate PC host, can perform additional tests to verify or debug any memory issues.

10 Additionally, as previously discussed, a user can access transmitter and receiver logic of the link interfaces 210, 212, which allows for control over the memory bus to be obtained. Such control can be used to monitor interface calibration or be used as a manual override for calibration when desired. As a result, a user can potentially debug the memory bus without a high speed logic analyzer and scope. Moreover, the transmitters and  
15 receivers of the memory interfaces 270 and the link interfaces 210, 212 can be monitored or driven through the maintenance bus 296 and diagnostic engine 290 by using debug software. For example, the transmitter drive strength and slew rate can be controlled. The receiver filter coefficients and clock placement can also be controlled. For the purpose of performing diagnostics, resulting evaluation data can be fed to a graphical user interface to  
20 display signal quality.

The DMA engine 286 can also be used for running diagnostics in the system. For example, known good data patterns can be loaded in memory of the memory hub 200, or known good system memory, and used to test the system memory. In alternative embodiments of the present invention, the DMA engine 286 is further capable  
25 of generating desired memory bus or link access patterns. Additionally, the DMA engine 286 has random request spacing, address and data pattern capability, and can be operated in a very controlled or a random mode. The DMA engine 286 can be delegated the responsibility for generating requests to the memory controllers 280 or link interfaces 210,

212 and checking the return data. The DMA engine 286 can further include various data buffers to store trace history, which can be accessed through the maintenance bus 296 and the diagnostic engine 290 to be evaluated by the user.

Figure 3 illustrates a diagnostic engine 300 according to an embodiment of the present invention. The diagnostic engine 300 can be substituted for the diagnostic engine 296 of Figure 2. It will be appreciated that Figure 3 is a functional block diagram representative of a suitable diagnostic engine, and is not intended to limit the scope of the present invention. The functional blocks shown in Figure 3 are conventional, and can be implemented using well known techniques and circuitry. It will be further appreciated that control signals and other functional blocks have been omitted from Figure 3 in order to avoid unnecessarily obscuring the present invention, and that the description provided herein is sufficient to enable those ordinarily skilled in the art to practice the invention.

Included in the diagnostic engine 300 is a diagnostic and BIST module 304 coupled to the maintenance bus 296 through a maintenance bus interface 306, through which the diagnostic engine 300 receives command and data signals from a user and through which the results of diagnostic testing can be accessed. A pattern generator 310 and sequencer 312 are coupled to the diagnostic and BIST module 304 for generating test patterns used for testing and diagnostic analysis and for translating commands provided to the diagnostic and BIST module 304 into memory commands applied to the system memory 240 (Figure 2). Further coupled to the diagnostic and BIST module 304, as well as to the pattern generator 310 and the sequencer 312, are link interface controller 320, memory interface controller 322, and DMA controller 324. Each of the controllers 320, 322, 324 are coupled to the switch 260 through the diagnostic bus 292.

In operation, the diagnostic and BIST module 304 receives command and data signals through the maintenance bus 296 from a user. In response, the diagnostic and BIST module 304 generates control signals and forwards the user supplied command and data signals to carry out the commands of the user. For example, the diagnostic and BIST module 304 may invoke the pattern generator 310 to begin generating a test pattern in

accordance with the user's commands and data, and also forward the user provided memory commands to the sequencer for translation into control signals that will be applied to the system memory 240 to carry out diagnostic memory operations. Based on the type of commands and data provided by the user, that is, the type of testing or diagnostic that will be performed, control signals are provided over the diagnostic bus 292 to the switch 260 and onto the appropriate memory hub functional blocks using the controllers 320, 322, 324. For example, as previously described, a user can monitor the link interface calibration and manually override the calibration by providing commands to the diagnostic engine through the maintenance bus 296. In such an instance, the diagnostic and BIST module 304 receives the user provided commands, and accesses the specified link interfaces 210, 212 through the link interface controller 320 and the switch 260 to monitor and adjust the link interface calibration.

It will be appreciated that the previous description of the memory hub 200 (Figure 2) and the diagnostic engine 300 (Figure 3) have been provided by way of example, and modifications to the memory hub 200 and the diagnostic engine can be made without departing from the scope of the present invention. For example, the previously described embodiments of the memory hub 200 includes a DMA engine 286. However, in alternative embodiments, a DMA engine is not present in the memory hub, and memory operations are performed under the command of the processor 104 (Figure 1) or the memory hub controller 128 instead. In another embodiment, the diagnostic engine 296 further includes self-testing and repair capabilities, such as those described in commonly assigned, co-pending U.S. Patent Application No. 10/222,393, entitled SYSTEM AND METHOD FOR SELF-TESTING AND REPAIR OF MEMORY MODULES, filed August 16, 2002, which is incorporated herein by reference.

From the foregoing it will be appreciated that, although specific embodiments of the invention have been described herein for purposes of illustration, various modifications may be made without deviating from the spirit and scope of the invention. Accordingly, the invention is not limited except as by the appended claims.